Modal Documentation

The Modal component is a powerful overlay system that extends the dialog service to provide custom modal dialogs with full content projection capabilities. It allows developers to create sophisticated modal interfaces with custom content, layouts, and interactions while maintaining consistent styling and behavior.
The modal system is built on top of the AavaDialogService and provides a flexible foundation for creating various types of modal dialogs, from simple confirmations to complex forms and content displays.

How to use

Modal Variants

```
<div class="con">
  <div class="col-12 col-sm-auto d-flex justify-content-center">
    <aava-button
      label=" Simple Modal"
      variant="secondary"
      (userClick)="openProperSimpleModal()"
    ></aava-button>
    <aava-button
      label=" Feedback Modal"
      variant="secondary"
      (userClick)="openProperFeedbackModal()"
    ></aava-button>
    <aava-button
      label=" Scrollable Modal"
      variant="secondary"
      (userClick)="openProperScrollableModal()"
    ></aava-button>
  </div>
</div>


---

  constructor(private dialogService: AavaDialogService) {}

 openProperSimpleModal() {
    this.dialogService.openModal(ProperSimpleModalComponent, {
      width: '400px',
    });
  }

  openProperFeedbackModal() {
    this.dialogService.openModal(ProperFeedbackModalComponent, {
      width: '400px',
    });
  }

  openProperScrollableModal() {
    this.dialogService.openModal(ProperScrollableModalComponent, {
      width: '484px',
    });
  }


---

import { Component } from "@angular/core";
import { AavaButtonComponent, AavaDialogService } from "play-comp-library";

@Component({
  selector: "app-proper-simple-modal",
  standalone: true,
  imports: [AavaButtonComponent],
  template: `
    <div style="padding:24px">
      <div class="ava-modal" style="border-radius:8px;">
        <div
          dialog-header
          style="display: flex;
      align-items: center;
```

```
gap: 12px;
align-self: stretch;"
  >
    <svg
      xmlns="http://www.w3.org/2000/svg"
      width="24"
      height="24"
      viewBox="0 0 24 24"
      fill="none"
    >
      <path
        d="M22 12C22 17.5228 17.5228 22 12 22M22 12C22 6.47715 17.5228 2 12 2M22 12H2M12 22C6.
        stroke="black"
        stroke-width="1.5"
        stroke-linecap="round"
        stroke-linejoin="round"
      />
    </svg>
    <h4
      style="margin-top:0;
  margin-bottom:0;
  line-height:0;
  color:#3B3F46;
  font-family: Mulish;
  font-size:24px;
  font-style: normal;
  font-weight: 700;
  line-height:28px;"
    >
      Heading
    </h4>
  </div>
  <div dialog-body>
    <p
      style="align-self: stretch;
color: #6B7280;
font-family: Inter;
font-size:16px;
font-style: normal;
font-weight: 400;
line-height:20px;
margin:24px 0px;"
    >
      This agent will be sent back for corrections and modifications.
      Kindly comment what needs to be done.
    </p>
  </div>
  <div dialog-footer style="display:flex; gap:12px">
    <aava-button
      label="Label"
      variant="secondary"
      size="md"
      (userClick)="onClose()"
      height="52px"
      width="165px"
    ></aava-button>
    <aava-button
      label="Label"
      variant="primary"
      size="md"
```

```
            (userClick)="onClose()"
            height="52px"
            width="165px"
          ></aava-button>
        </div>
      </div>
    </div>
  `,
})
export class ProperSimpleModalComponent {
  constructor(private dialogService: AavaDialogService) {}
  onClose() {
    this.dialogService.close();
  }
}


---

import { Component } from "@angular/core";
import {
  AavaButtonComponent,
  AavaDialogService,
  AavaCheckboxComponent,
} from "play-comp-library";

@Component({
  selector: "app-proper-scrollable-modal",
  standalone: true,
  imports: [AavaButtonComponent, AavaCheckboxComponent],
  template: `
    <div style="padding:24px">
      <div
        style="display:flex;
    flex-direction: column;
    gap:24px;"
      >
        <div
          dialog-header
          style="
    color: #3B3F46;
    font-family: Mulish;
    font-size: 24px;
    font-style: normal;
    font-weight: 700;
    line-height: 28px;
    justify-content:flex-start;
    "
        >
          <h3 style="margin-bottom:0px;">Heading</h3>
        </div>
        <div dialog-body>
          <div>
            <h4>Heading small</h4>
            <p>
              This agent will be sent back for corrections and modifications.
              Kindly comment what needs to be done.
            </p>
            <h4 style="margin: 0 0 8px 0; font-size: 14px;">Heading small</h4>
            <p style="margin: 0 0 16px 0; color: #666; font-size: 14px;">
```

```
          This agent will be sent back for corrections and modifications.
          Kindly comment what needs to be done.
        </p>
        <h4>Heading small</h4>
        <p>
          This agent will be sent back for corrections and modifications.
          Kindly comment what needs to be done.
        </p>
        <h4>Heading small</h4>
        <p>
          This agent will be sent back for corrections and modifications.
          Kindly comment what needs to be done.
        </p>
        <div>
          <h4>Heading small</h4>
          <p>
            This agent will be sent back for corrections and modifications.
            Kindly comment what needs to be done.
          </p>
        </div>
        <div
          style="
  display: flex;
  height: 24px;
  padding: 0 0;
  align-items: center;
  gap: 8px;
  align-self: stretch;"
        >
          <aava-checkbox variant="default" size="sm"></aava-checkbox>
          <label> Accept Terms and Conditions</label>
        </div>
      </div>
    </div>
    <div
      dialog-footer
      style="
display:flex;
justify-content: flex-end;
flex-direction: row;
gap: 10px;
align-items: flex-end;
align-self: stretch;"
    >
      <aava-button
        label="Label"
        variant="secondary"
        size="md"
        (userClick)="onClose()"
      ></aava-button>
      <aava-button
        label="Label"
        variant="primary"
        size="md"
        (userClick)="onClose()"
      ></aava-button>
    </div>
  </div>
</div>
`,
```

```
})
export class ProperScrollableModalComponent {
  constructor(private dialogService: AavaDialogService) {}
  onClose() {
    this.dialogService.close();
  }
}


---

import { Component } from "@angular/core";
import {
  AavaButtonComponent,
  AavaDialogService,
  AavaIconComponent,
  AavaTextboxComponent,
} from "play-comp-library";

@Component({
  selector: "app-proper-feedback-modal",
  standalone: true,
  imports: [AavaButtonComponent, AavaIconComponent, AavaTextboxComponent],
  template: `
    <div style="padding:24px">
      <div dialog-header>
        <h3
          style="color: #3B3F46;
        font-family: Mulish;
        font-size: 24px;
        font-style: normal;
        font-weight: 700;
        line-height:28px;"
        >
          Heading
        </h3>
      </div>
      <div dialog-body>
        <div style="text-align: center; margin:24px 0px">
          <aava-icon
            iconName="circle-check"
            iconSize="70px"
            iconColor="green"
          ></aava-icon>
        </div>
        <p
          style="align-self: stretch;
        color: #6B7280;
        text-align: center;
        font-family: Inter;
        font-size: 16px;
        font-style: normal;
        font-weight: 400;
        line-height: 20px;
        margin-bottom:24px"
        >
          This agent will be sent back for corrections and modifications. Kindly
          comment what needs to be done
        </p>
        <div style="margin-bottom:24px">
```

```
            <aava-textbox
              label="Feedback Input"
              variant="default"
              placeholder="Enter text here"
              size="xl"
            ></aava-textbox>
          </div>
        </div>
        <div dialog-footer style="display:flex; gap:12px">
          <aava-button
            label="Label"
            variant="secondary"
            size="md"
            (userClick)="onClose()"
            height="52px"
            width="165px"
          ></aava-button>
          <aava-button
            label="Label"
            variant="primary"
            size="md"
            height="52px"
            (userClick)="onClose()"
            width="165px"
          ></aava-button>
        </div>
      </div>
    `,
})
export class ProperFeedbackModalComponent {
  constructor(private dialogService: AavaDialogService) {}
  onClose() {
    this.dialogService.close();
  }
}
```

Simple Modal Features

• Header Section : Custom header with icon and title
• Body Content : Descriptive text content
• Footer Actions : Primary and secondary action buttons
• Responsive Design : Adapts to different screen sizes
• Accessibility : Proper ARIA attributes and keyboard navigation

Feedback Modal Features

• Visual Feedback : Success icon with color-coded styling
• Input Fields : Text input for user feedback
• Centered Layout : Optimized for form interactions
• Action Buttons : Clear primary and secondary actions
• User Experience : Intuitive feedback collection flow

Scrollable Modal Features

• Content Scrolling : Handles overflow content gracefully
• Form Elements : Checkbox and form controls
• Structured Content : Organized sections with headings
• Flexible Height : Adapts to content length
• Footer Positioning : Actions remain accessible during scroll

Features

Content Projection

The modal system uses Angular's content projection to provide flexible content structure:
• [dialog-header] : Header section for titles, icons, and navigation
• [dialog-body] : Main content area for forms, text, or components
• `[dialog-footer] : Footer section for action buttons and controls

Flexible Sizing

• Width Control : Customizable width with responsive constraints
• Height Management : Automatic height adjustment or fixed heights
• Responsive Behavior : Adapts to different screen sizes
• Max Dimensions : Configurable maximum width and height

Service Integration

• DialogService : Centralized modal management
• Component Injection : Dynamic component rendering
• Configuration Options : Flexible modal configuration
• Lifecycle Management : Proper cleanup and resource management

Accessibility

• Keyboard Navigation : Full keyboard support (Tab, Escape, Enter)
• Screen Reader : Proper ARIA attributes and labels
• Focus Management : Focus trapping and restoration
• High Contrast : Maintains accessibility standards

API Reference

AavaDialogService.openModal()

ModalConfig Interface

Property	Type	Default	Description
width	string	-	Modal width (e.g., '600px')
maxWidth	string	-	Maximum width constraint

Property	Type	Default	Description
height	string	-	Modal height
maxHeight	string	-	Maximum height constraint
showCloseButton	boolean	true	Show close button
backdrop	boolean	true	Show backdrop overlay
closeOnBackdrop	boolean	true	Close on backdrop click

Content Projection Attributes

Attribute	Description
[dialog-header]	Header section for modal title and navigation
[dialog-body]	Main content area for modal content
`[dialog-footer]	Footer section for action buttons

Best Practices

Modal Design

• Clear Purpose : Make modal purpose obvious through title and content
• Appropriate Sizing : Choose modal sizes that fit content without overwhelming
• Consistent Layout : Use consistent header, body, and footer structure
• Visual Hierarchy : Maintain clear visual hierarchy within the modal
• Responsive Design : Ensure modals work well on all screen sizes

Content Organization

• Header Content : Include clear titles and optional icons
• Body Structure : Organize content logically with proper spacing
• Footer Actions : Place primary actions on the right, secondary on the left
• Content Length : Keep content concise or implement scrolling for long content
• Form Elements : Use appropriate form controls and validation

User Experience

• Clear Actions : Make button actions and consequences clear
• Escape Options : Always provide a way to close the modal
• Loading States : Show appropriate loading indicators for async operations
• Error Handling : Provide clear error messages and recovery options
• Accessibility : Ensure keyboard navigation and screen reader support

Performance

• Lazy Loading : Load modal content only when needed
• Component Reuse : Reuse modal components when possible
• Memory Management : Properly clean up modal instances
• Change Detection : Use OnPush strategy for optimal performance
• Bundle Optimization : Import only needed modal components

Styling

The modal component uses CSS custom properties for theming: